

CS & ICT Departments

7th Semester

Operating System Concepts

Chapter 1: Introduction

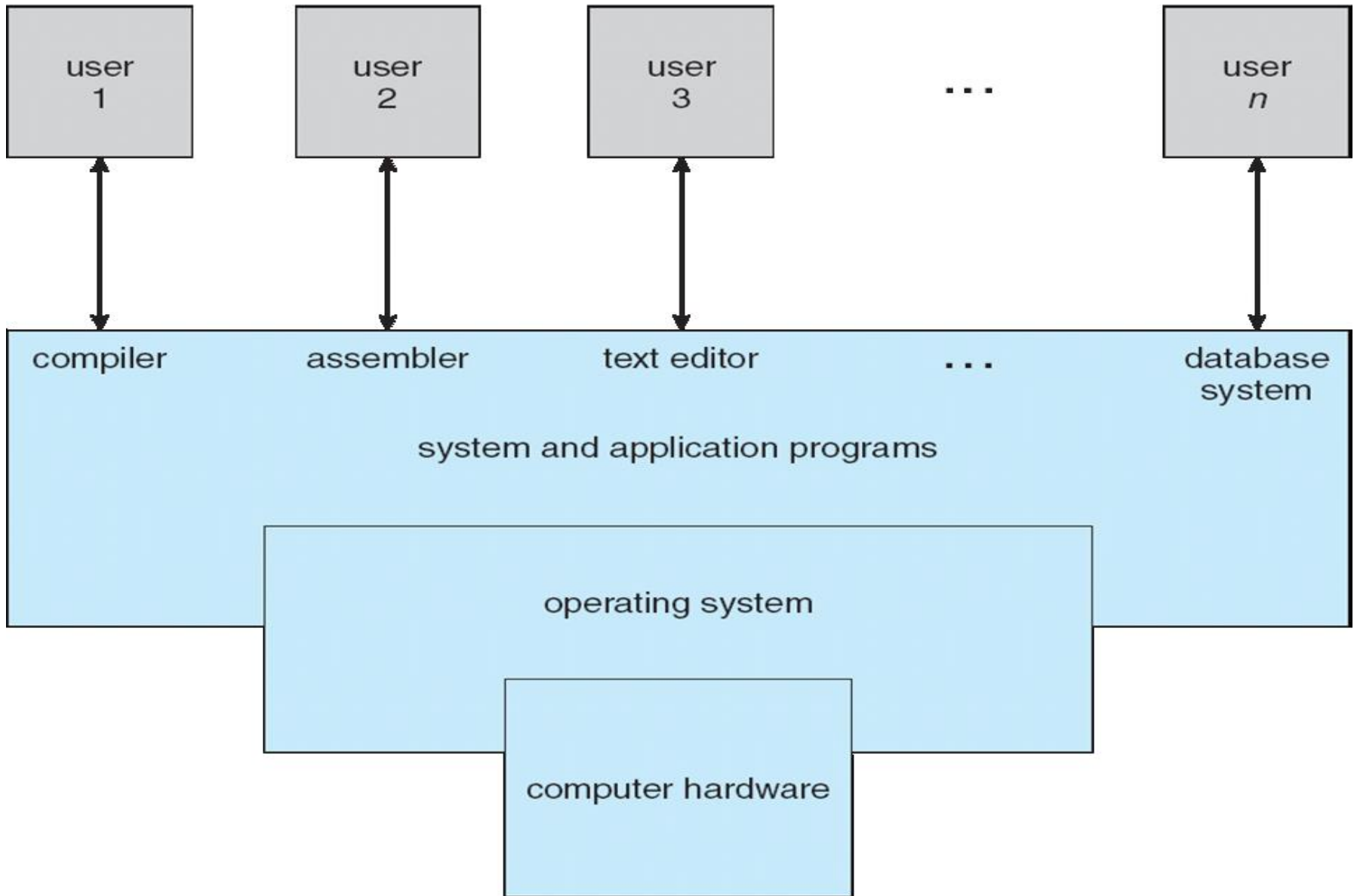
What is an Operating System?

- A program that acts as an intermediary between a user of a computer and the computer hardware
- Operating system goals:
 - Execute user programs and make solving user problems easier
 - Make the computer system convenient to use
 - Use the computer hardware in an efficient manner

Computer System Structure

- Computer system can be divided into four components:
 - Hardware – provides basic computing resources
 - ▶ CPU, memory, I/O devices
 - Operating system
 - ▶ Controls and coordinates use of hardware among various applications and users
 - Application programs – define the ways in which the system resources are used to solve the computing problems of the users
 - ▶ Word processors, compilers, web browsers, database systems, video games
 - Users
 - ▶ People, machines, other computers

Four Components of a Computer System



What Operating Systems Do

□ Depends on the point of view

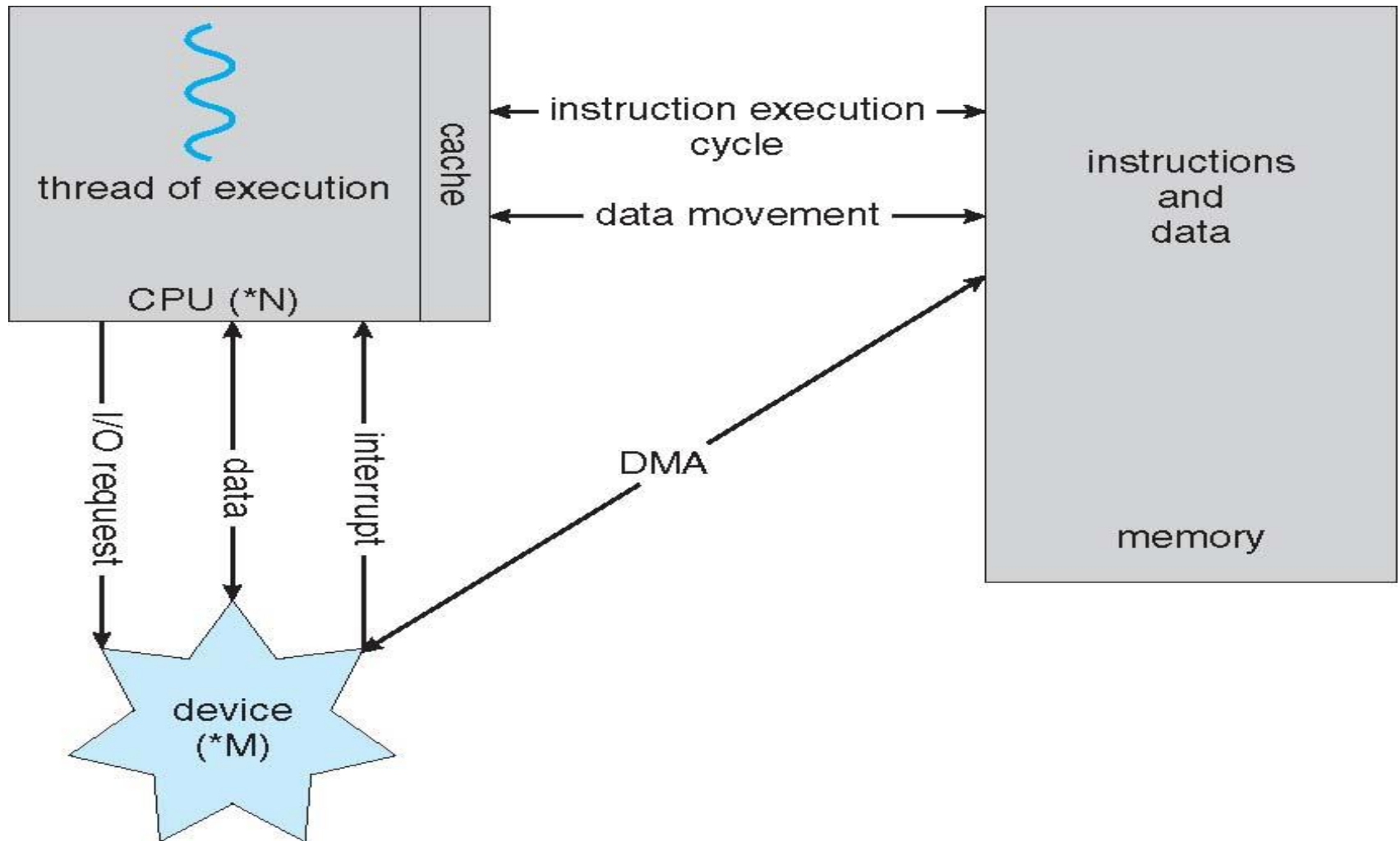
□ User view:

- ▶ Users want convenience, **ease of use** and **good performance**
 - Don't care about **resource utilization**
- ▶ But shared computer such as **mainframe** or **minicomputer** must keep all users happy
- ▶ Users of dedicate systems such as **workstations** have dedicated resources but frequently use shared resources from **servers**
- ▶ Handheld computers are resource poor, optimized for usability and battery life
- ▶ Some computers have little or no user interface, such as embedded computers in devices and automobiles

What Operating Systems Do

- System view:
 - OS is a **resource allocator**
 - ▶ Manages all resources
 - ▶ Decides between conflicting requests for efficient and fair resource use
 - OS is a **control program**
 - ▶ Controls execution of programs to prevent errors and improper use of the computer

How a Modern Computer Works

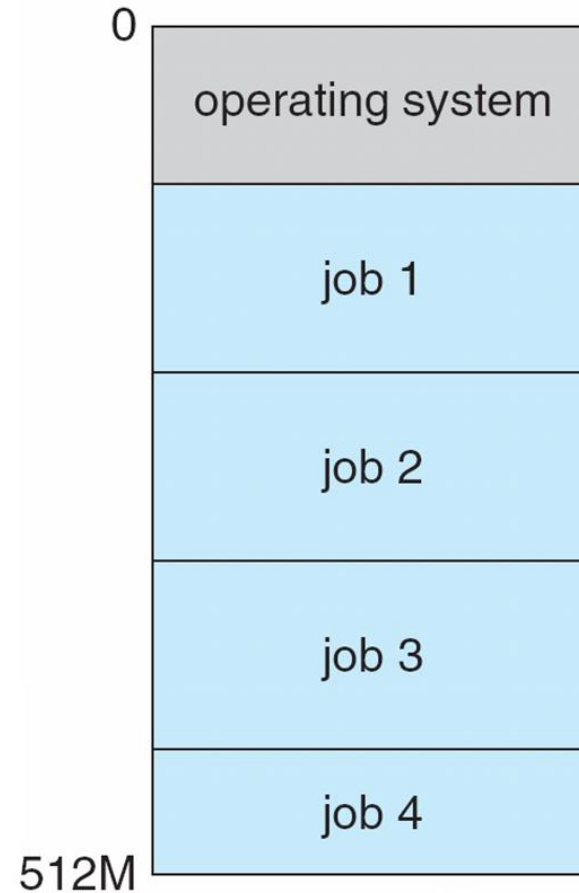


A von Neumann architecture

Operating System Structure

- **Multiprogramming (Batch system)** needed for efficiency
 - Single user cannot keep CPU and I/O devices busy at all times
 - Multiprogramming organizes jobs (code and data) so CPU always has one to execute
 - A subset of total jobs in system is kept in memory
 - One job selected and run via **job scheduling**
 - When it has to wait (for I/O for example), OS switches to another job
- **Timesharing (multitasking)** is logical extension in which CPU switches jobs so frequently that users can interact with each job while it is running, creating **interactive** computing
 - **Response time** should be < 1 second
 - Each user has at least one program executing in memory ⇒ **process**
 - If several jobs ready to run at the same time ⇒ **CPU scheduling**
 - If processes don't fit in memory, **swapping** moves them in and out to run
 - **Virtual memory** allows execution of processes not completely in memory

Memory Layout for Multiprogrammed System



Operating-System Operations

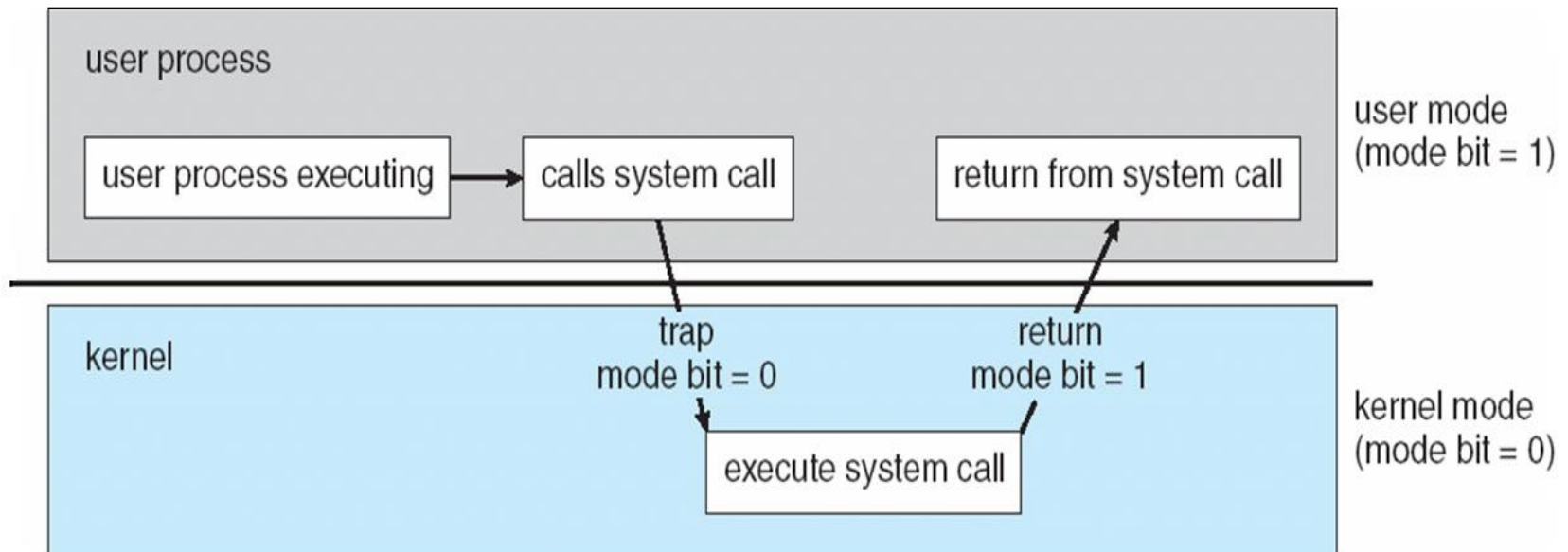
- **Interrupt driven** (hardware and software)
 - Hardware interrupt by one of the devices
 - Software interrupt (**exception** or **trap**):
 - ▶ Software error (e.g., division by zero)
 - ▶ Request for operating system service
 - ▶ Other process problems include infinite loop, processes modifying each other or the operating system

Operating-System Operations (cont.)

- **Dual-mode** operation allows OS to protect itself and other system components
 - **User mode** and **kernel mode** (also called supervisor mode, system mode or privileged mode)
 - **Mode bit** provided by hardware
 - ▶ Provides ability to distinguish when system is running user code or kernel code
 - ▶ Some instructions designated as **privileged**, only executable in kernel mode
 - ▶ System call changes mode to kernel, return from call resets it to user
- Increasingly CPUs support multi-mode operations
 - i.e. **virtual machine manager (VMM)** mode for guest **VMs**

Transition from User to Kernel Mode

- Timer to prevent infinite loop / process hogging resources
 - Timer is set to interrupt the computer after some time period
 - Keep a counter that is decremented by the physical clock.
 - Operating system set the counter (privileged instruction)
 - When counter zero generate an interrupt
 - Set up before scheduling process to regain control or terminate program that exceeds allotted time



Process Management

- ❑ A process is a program in execution. It is a unit of work within the system. Program is a *passive entity*, process is an *active entity*.
- ❑ Process needs resources to accomplish its task
 - ❑ CPU, memory, I/O, files
 - ❑ Initialization data
- ❑ Process termination requires reclaim of any reusable resources
- ❑ Single-threaded process has one **program counter** specifying location of next instruction to execute
 - ❑ Process executes instructions sequentially, one at a time, until completion
- ❑ Multi-threaded process has one program counter per thread
- ❑ Typically system has many processes, some user, some operating system running concurrently on one or more CPUs
 - ❑ Concurrency by multiplexing the CPUs among the processes / threads

Process Management Activities

The operating system is responsible for the following activities in connection with process management:

- ❑ Creating and deleting both user and system processes
- ❑ Suspending and resuming processes
- ❑ Providing mechanisms for process synchronization
- ❑ Providing mechanisms for process communication
- ❑ Providing mechanisms for deadlock handling

Memory Management

- ❑ To execute a program all (or part) of the instructions must be in memory
- ❑ All (or part) of the data that is needed by the program must be in memory.
- ❑ Memory management determines what is in memory and when
 - ❑ Optimizing CPU utilization and computer response to users
- ❑ Memory management activities
 - ❑ Keeping track of which parts of memory are currently being used and by whom
 - ❑ Deciding which processes (or parts thereof) and data to move into and out of memory
 - ❑ Allocating and deallocating memory space as needed

Storage Management

- OS provides uniform, logical view of information storage
 - Abstracts physical properties to logical storage unit - **file**
 - Each medium is controlled by device (i.e., disk drive, tape drive)
 - ▶ Varying properties include access speed, capacity, data-transfer rate, access method (sequential or random)

- File-System management
 - Files usually organized into directories
 - Access control on most systems to determine who can access what
 - OS activities include
 - ▶ Creating and deleting files and directories
 - ▶ Primitives to manipulate files and directories
 - ▶ Mapping files onto secondary storage
 - ▶ Backup files onto stable (non-volatile) storage media

Mass-Storage Management

- Usually disks used to store data that does not fit in main memory or data that must be kept for a “long” period of time
- Proper management is of central importance
- Entire speed of computer operation depend on disk subsystem and its algorithms
- OS activities
 - Free-space management
 - Storage allocation
 - Disk scheduling
- Some storage need not be fast
 - Tertiary storage includes optical storage, magnetic tape
 - Still must be managed – by OS or applications
 - Varies between WORM (write-once, read-many-times) and RW (read-write)

I/O Subsystem

- One purpose of OS is to hide properties of hardware devices from the user
- I/O subsystem responsible for
 - Memory management of I/O including buffering (storing data temporarily while it is being transferred), caching (storing parts of data in faster storage for performance), spooling (the overlapping of output of one job with input of other jobs)
 - General device-driver interface
 - Drivers for specific hardware devices